Implementing Strategic Planning Capabilities within the Mars Relay Operations Service

Franklin Hy[1], Roy Gladden[2], Dan Allard[3], and Michael Wallick[4]
Jet Propulsion Laboratory, California Institute of Technology Pasadena, CA 91109

Since the Mars Exploration Rovers (MER), Spirit and Opportunity, began their travels across the Martian surface in January of 2004, orbiting spacecraft such as the Mars 2001 Odyssey orbiter have relayed the majority of their collected scientific and operational data to and from Earth. From the beginning of those missions, it was evident that using orbiters to relay data to and from the surface of Mars was a vastly more efficient communications strategy in terms of power consumption and bandwidth compared to direct-to-Earth means. However, the coordination between the various spacecraft, which are largely managed independently and on differing commanding timelines, has always proven to be a challenge. Until recently, the ground operators of all these spacecraft have coordinated the movement of data through this network using a collection of ad hoc human interfaces and various, independent software tools. The Mars Relay Operations Service (MaROS) has been developed to manage the evolving needs of the Mars relay network, and specifically to standardize and integrate the relay planning and coordination data into a centralized infrastructure. This paper explores the journey of developing the MaROS system, from inception to delivery and acceptance by the Mars mission users.

Nomenclature

Ace	=	Call sign for Flight Project Mission Controllers
API	=	Application Programming Interface
CSF	=	Contact Schedule File
DOM	=	Distributed Object Manager
LOPTG	=	Lander Orbit Propogation and Timing Geometry
MaROS	=	Mars Relay Operations Service
MER	=	Mars Exploration Rovers
MGSS	=	Multi-mission Ground Systems and Services
MRO	=	Mars Reconnaissance Orbiter
OAF	=	Orbiter Acknowledgement File
ORF	=	Orbiter Request File
OSOE	=	Orbiter Sequence of Events
ReST	=	Representational State Transfer

[1] Computer Scientist, Multimission Ground Data System Engineering, 264-214.
[2] Multimission Relay Operations Lead, Planning and Execution System Engineering, 264-235.
[3] Computer Scientist, Distributed Systems Technologies, 301-480.
[4] Computer Scientist, Operations Planning Software Group, 301-250D.

I. Introduction

Relaying data between spacecraft is often mistaken as a simple technical challenge, isolated to the development of communications protocols. While it is true that the chaining together of communications links between the various components of a relay network is a critical activity, this is not the only aspect of the problem that must be addressed before a functional relay network can be instantiated. The technical infrastructure must also be complemented by a robust planning and coordination effort which crisply defines opportunities and means when data can be transferred between each node in the network.

Generally, the relay coordination process can be described as including five different functions:

- **Strategic Planning Process:** A planning and coordination effort to define opportunities for transferring information from one node in the network to another. This typically includes preparing each node to participate by scheduling communications sessions with ground stations and by generating command sequences for the related spacecraft.
- **Forward Link Commanding Process:** A mechanism to transfer forward-link data destined for the end-point spacecraft, in this case a lander or rover on the surface of Mars. This implies a means to acquire and properly format transfer data products from landed vehicle's operators by each intermediate node in the network. Additionally, this data must be appropriately scheduled for transfer through the network.
- **Return-Link Data Flow Process:** A mechanism to transfer return-link data destined for the operators of an end-point asset, in this case the operators of a lander or rover on the surface of Mars. This implies a means to package and properly handle any data that may have been acquired by intermediate assets, including methods to manage incomplete data sets. Returning this data to the users must typically be a low-latency activity.
- **Relay Accountability Process:** An approach to monitoring the behavior of the relay network. This implies a need to understand the performance of each link in the network, with appropriate reporting mechanisms.
- **Tactical Planning Process:** An ability to respond to changing conditions within the network, as the conditions of each node in the network may change at any point in the planning and implementation timeline. For example, a lander on the surface of Mars may experience environmental effects that make it difficult to adhere to previously planned and scheduled relay activities. In this case, the network must provide a robust architecture for managing these last minute changes.

Since before the landing of the twin Mars Exploration Rovers, these five functions were managed as independent problems, solved individually and sometimes as their needs became apparent. This condition led to an approach to managing the relay network that involved separate and distinct software mechanisms, and in many cases processes that required human involvement.

A. Challenges

The Mars relay network is comprised of spacecraft that are each, foremost, platforms for scientific discovery. Particularly in the case of the orbiters, this often implies that there is a direct conflict between acquiring scientific data and supporting relay activities. Therefore, each participant in the relay network imposes certain restrictions on how and when relay activities can be performed. This policy-making is a very human activity and requires cooperation between those who manage each node in the network; the implementation and execution of relay activities must be responsive to this.

In addition, each node in the current network was not engineered specifically to perform relay activities. None of the current generation of orbiters were designed in such a way that they could be receptive to signals from a landed asset at any time; they must be directly commanded to initiate relay sessions. This commanding is implemented on the operational timeline of each of the orbiters, typically spanning weeks of planning and implementation for command sequences that are also weeks in duration. This is in contrast to the typical planning horizon of the landed assets, which must be responsive to changing environmental conditions on the surface of Mars and is therefore no longer than several days. The matching up of these commanding timelines can be a challenge to accommodate for relay planning purposes.

With each node being managed independently, along with differing ground system and onboard implementations, finding mechanisms to correlate the relay plan across the entire planning horizon is critical. Understanding the most recently implemented state of any particular communications session can be essential to the ongoing operations of the landed assets, which may need to respond quickly to unforeseen circumstances.

B. Mars Relay Operations Service(MaROS)

To rectify the key deficiencies of the legacy processes and tools, NASA's Mars Program Office and the Jet Propulsion Laboratory's Multi-Mission Ground Systems and Services (MGSS) Program have funded the implementation of a new system to support this network, called the Mars Relay Operations Service, or MaROS. This system standardizes the interfaces between current and future participants in the relay network, and provides an infrastructure to store and retrieve all relay-related planning and operations data. By centralizing this information in a secure database and by providing appropriate, standardized interfaces, automation can be developed to reduce the cost and complexity of performing relay operations. MaROS is being developed and deployed operationally in phases. We describe the first three phases of development below. Future phase development will be defined as necessary when issues are identified within or following the first three phases.

1. Phase 1

There are a variety of different data types that must be correlated to support the relay planning processes, as illustrated in Fig. 1. In addition to the navigation data of the orbiters, which is required to understand geometric view periods between the orbiters and the landed asset; additional data such as uplink and downlink opportunities to those orbiters from Earth is required. From this, an understanding of the latencies involved in transferring data through the network from landed asset ground operators to the landed asset can be derived, along with a clear understanding of any operational constraints that may need to be accounted for when planning relay activities.

The first phase in the development of MaROS focused on this planning activity, specifically addressing the Strategic Planning Process. This development was deployed for use in mission operations in mid-2010 for use by the Mars Relay Network. In this phase, the users were provided a read-only, Adobe Flash Player-based web interface to visualize the network's planning data, and a set of command-line utilities to interact with the service. Server-side interfaces were built to support both the web and command-line interactions. This first phase also put into place standardized mechanisms to report the results of relay sessions between a lander and an orbiter. This addresses the Relay Accountability Process.

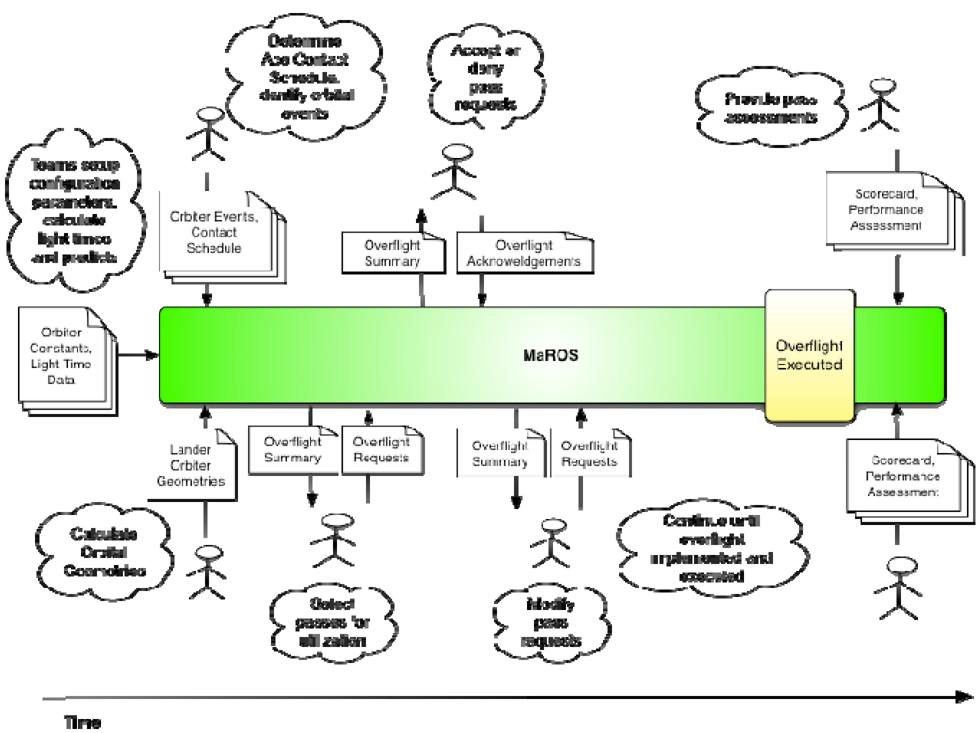

Figure 1. MaROS Data Types. *Collection of data types input into MaROS and how they are used by the lander and orbiter teams.*

One of the primary objectives of MaROS is to modernize the interfaces that were originally put into place to support the Mars Exploration Rovers, Spirit and Opportunity. These previous interfaces, involving human interactions in both email and by phone calls, have been replaced by software mechanisms which make common the interactions between the various operators, removing much of the ambiguity and greatly reducing the potential for error. This is particular important in the Tactical Planning Process when decisions must be made quickly and reliably, as will be discussed shortly.

2. Phase 2

The second phase of development of MaROS focused on the Forward-Link Commanding Process. This phase of MaROS was deployed for use in mission operations in early 2011 for use by the Mars Relay Network.With the previous architecture, the landed asset ground operators were required to construct an interface directly with the orbiter ground operators in order to pass forward-link command products through the network. This involved different processes and different interfaces for each orbiter, which needed to be put into place for each lander. With an evolving relay network which continues to introduce new vehicles, this exponential increase of interfaces is costly, time-consuming, and can ultimately be the source of errors due to the disparate processes.

MaROS addresses this issue by serving as an intermediary in the Forward-Link Commanding Process. Landed assets can stage forward-link command products within the MaROS system and submit them directly through MaROS to the orbiting asset using the same infrastructure and interfaces. In this way, the lander projects need only develop the interface to MaROS and they gain access to any orbiter that is participating in the relay network. Similarly, the orbiter projects need only develop their interface to MaROS and they can provide relay services to any lander. This greatly simplifies the implementation costs, streamlines the operational processes, and reduces the potential for commanding errors during regular mission operations.

3. Phase 3

The third phase of the MaROS development is focused on the Tactical Planning Process. This architecture requires a much higher level of interaction between the participating projects, and necessitates the standardization of interfaces for communicating a wide variety of data between them. This third phase of MaROS is anticipated to be fully deployed for use by the Mars relay network before the end of 2011.

For example, at the strategic level, a lander project may have requested that a relay session be performed between the lander and an orbiter at a specific data rate. This data rate would generally be implemented by the orbiter project in a relatively long-term command product that executes on the commanding timeline of the orbiter. However, due to conditions on the surface of Mars, it may be necessary for a lander project to have that data rate changed. If able, the orbiter project would be requested to modify the data rate for the relay session from that which was previously planned. The orbiter project may have a variety of mechanisms to effect such a change, and these mechanisms would likely be unique to that orbiter. The lander project is likely not interested in the details of those mechanisms, and so would prefer to indicate the need to make the change in the most generic way possible. MaROS will support this by providing a standardized means to communicate this request to the orbiter.

MaROS supports this coordination by providing both a file-based mechanism for exchanging the information between the projects, and by making this data available through the web interface. MaROS also distributes notifications to subscribed users alerting them to these required changes. In the previous architecture, such a change would be requested by email or by a phone call, and there was always the potential for miscommunication or error when implementing it. With MaROS, the software-based interaction by its nature standardizes the discussion and ensures consistency in the interaction.

II. Architecture

The MaROS system has a variety of users from different missions, with different roles, interacting with the system on separate timelines. Users must access the system on a wide variety of systems ranging from dedicated workstations to personal mobile laptops from locations worldwide. To best meet this utilization profile MaROS is architected as a shared, centralized service. A set of thin-layer clients are provided to customers to facilitate the human and machine user interfaces; however, all data management is handled by the central service. Accessibility is a high priority and thus a web user interface is provided. A command line interface has also been provided to customers whose legacy machines may not support the minimum browser requirements to run the web interface.

Fig. 2 depicts the architecture of the system which will be discussed in the following section.

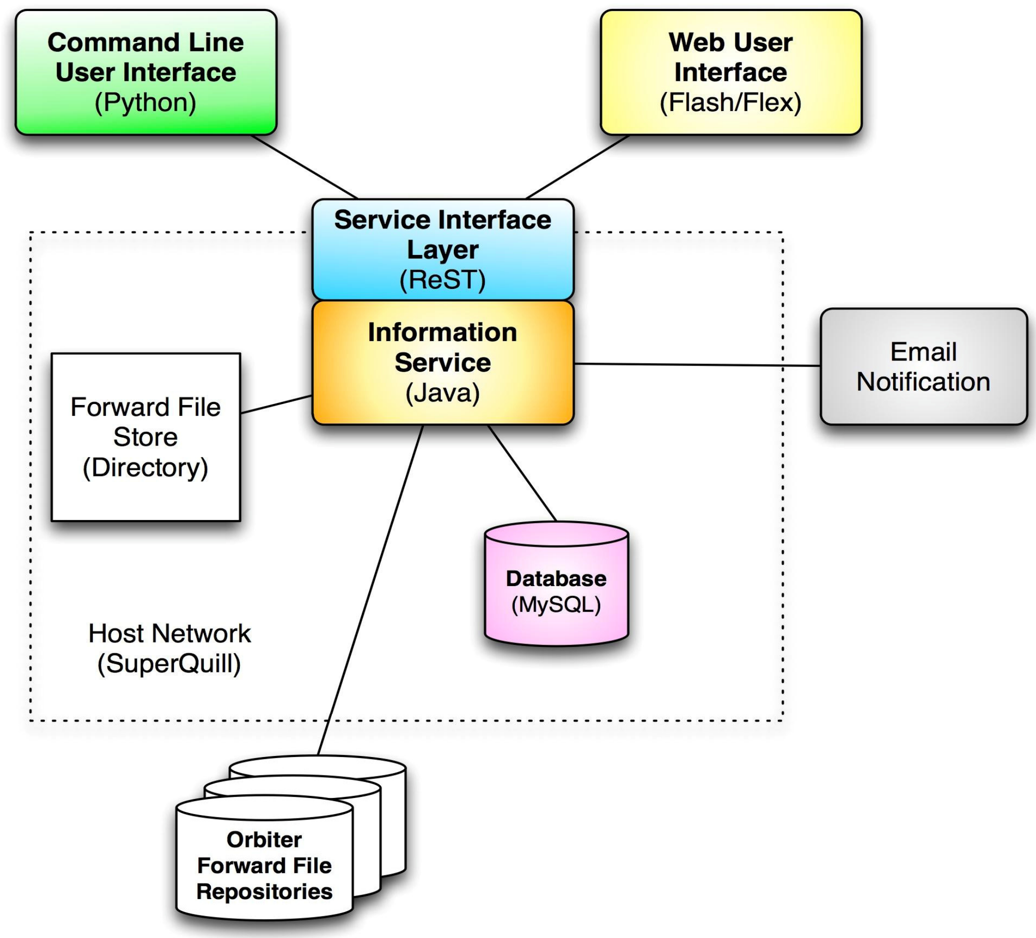

Figure 2. MaROS Architecture and Components.

A. Information Service

At the heart of the architecture is the central information service. All requests to the service are received via a service interface layer; in turn the information service performs all business logic functions for the system. Primary functions of the service include:

- Management of data uploaded to the service, including access permissions, validation of files, database interactions, and triggering of all derived functionality.
- Derivation of a variety of pass planning warnings and conflicts through the correlation of overlapping data sets.
- Derivation of forward- and return-link latencies, which are times associated with the delivery of data through the network both towards and from the landed spacecraft via an orbiting relay spacecraft.
- Delivery of lander forward-link products to an orbiter prior to being transferred to a lander during a designated relay opportunity.
- Delivery of notifications to end-users that specific types of events have occurred.

The web service is implemented as a Java application “ReSTlet”, exposing only the https interface to external clients. Note that while the service handles all requests to persist data or retrieve data from storage, the actual storage of data is managed by a relational database.

B. ReST

The information service provides access to information via Representational State Transfer (ReST) patterns. With these patterns, information is accessed as a set of "resources" at a specific URL, which are provided as a "representation" (e.g. XML, JSON) that is understood by both the client and server. For example, lander request information may be provided via a resource called "OverflightRequests", with additional name-value pair parameters provided to filter the desired request data. All external interfaces interact with the ReSTful service layer; no access is provided directly to any internal component such as the back end database.

The value of providing all services via ReST is significant. One primary benefit is that client applications may be written with any web-capable technology. For example, when the command line client was first being developed, we experimented with several different languages to implement it with including Perl, Java, and Python. Additionally, the ReST layer fully abstracts the internals of the system from external integration, meaning that the entire back-end could be retrofitted without any impact on external clients, assuming that the retrofit meets the specification of the original ReST interface. ReST patterns also come with a notion of *statelessness*, in that the server does not maintain information on the state of the client, which greatly simplifies the client-server architecture.

C. Database

All planning information is stored in a relational database. MaROS utilizes a MySQL database that is provided and maintained by the host network, however the software is designed in a manner that does not use any MySQL specific features.

D. Command Line User Interface

The command line user interface, which is provided to many users of the service, consists of several scripts which perform ReST transactions with MaROS. They are capable of uploading, downloading, and manipulating data received from or transferred to the information service. The scripts are designed to run on a Solaris 9 machine with Python 2.3 to provide support to legacy systems unable to upgrade their current operations environment.

It was required to keep each script as a single file to facilitate distribution of the scripts to several users. In those cases where there was a need to include an outside package, the needed logic was included within the script with comments blocks indicating the start and end of the package. It is fortuitous that nearly all of the business logic is performed by the information service, which minimizes the impact this single-script requirement imposes.

The command line scripts primarily validate that a user is providing valid inputs for the desired transaction, and then wraps their inputs to conform with the ReST Application Programming Interface (API) and performs the transaction with MaROS. The script is blind to the actions within the information service during the transaction and merely checks the response from that service as it returns simple status messages, which indicate whether to display error messages or to provide a success message or a data query. The command line scripts can be forced to output everything it receives prior to output filtering for extra debugging.

E. Web User Interface

The web component of MaROS provides users with an interactive visualization of the Mars network at any given time. The web interface is a flash program built using Adobe Flex. This means that MaROS is accessible to any user with a modern web browser and Internet connection. The information provided is gathered through the same ReSTful interfaces that are used via the command-line scripts. In addition to providing a graphical user interface for the service, it also relieves the user of being concerned about network interactions.

Once a user has logged into the web page, they are presented with a "navigation portal" that provides a means of accessing different areas of the MaROS web component. These tasks include (and are discussed below) displaying a visual timeline of the network, displaying extended information regarding an orbiter/lander relay session, interacting with lander forward-link products in the MaROS system for transfer to an orbiter system prior to a relay session, and performing user administration at the system and mission level. Additional capabilities are currently under development.

The visual timeline shows the state of the Mars Network across a specified time range. The default time is two hours prior to the present time through 22 hours after the present time, however any time range may be specified. Each spacecraft in the network (orbiter or lander) is shown in its own display, as shown in Fig. 3. The horizontal axis for both landers and orbiters represents time. In a lander's display, the Local Mars Time (LMT) is displayed. In an orbiter's display, the orbit number is shown instead. The exact time or orbit number of the mouse position is also shown to the user.

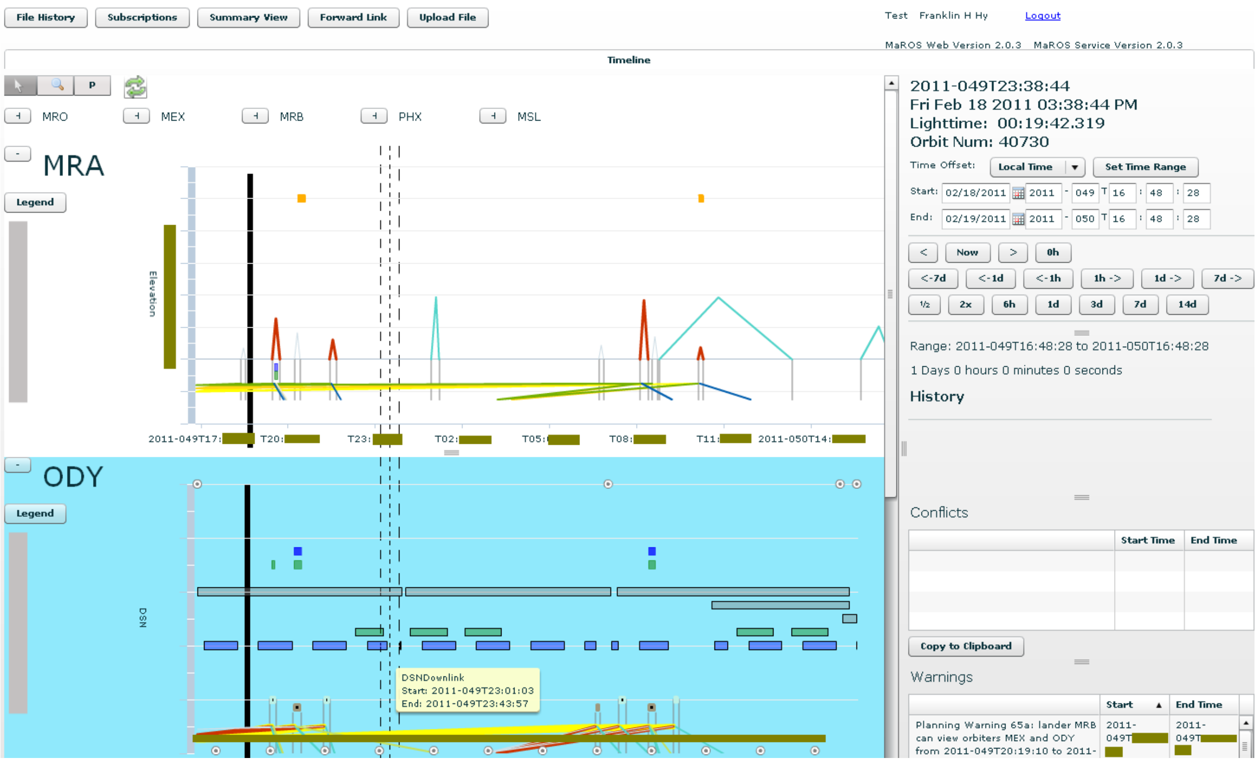

Figure 3. MaROS Web User Interface. *Orbiter and Lander visual timeline*

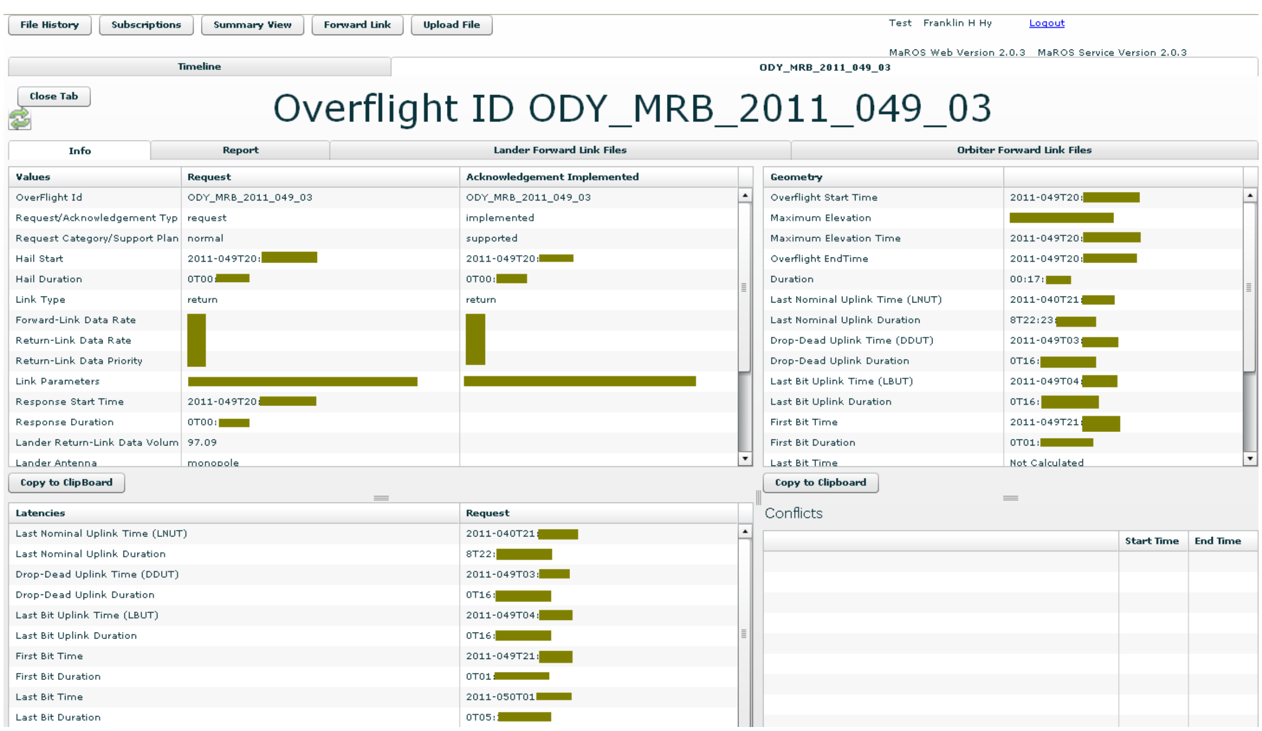

Figure 4. MaROS Web User Interface. *Sample Overflight Information*

The vertical axis of the lander timelines is used to represent the elevation angle of the orbiter as an overflight progresses. A lander's timeline also includes forward- and return-link latencies associated with each overflight as well as information regarding requested passes in view. The orbiter timeline shows the overflights, latencies associated with each overflight, and deep space antenna tracking coverage and other operator information.While the timeline provides basic information about a particular overflight, a user may select a specific overflight, either from the navigation portal or by clicking directly on it in the timeline. This opens a new page displaying more information about the selected overflight, as shown in Fig. 4. From this page, a user may see more in-depth statistics regarding that pass, request the overflight for a relay session, or specify forward-link products for transfer during the pass.

F. Email Notification

Email provides value as a "push" form of notification for most users of the system. MaROS provides email notifications for a wide range of system actions delivered to subscribed users on a mission-by-mission basis. Email notifications can inform one team that another remote team has completed a task such as submitting a new request for relay support, and thus respond with a follow-up activity in a timely fashion. The service application treats data activities as "events", generates a textual representation of these events and sends out individual email messages to subscribed users via the Java Mail API. Future versions of the system are planned to include different types of notifications, such as event notification via a Java Message Service (JMS) server and also through a "live feed" of data to the web user interface.

G. Forward File Store

One primary function of MaROS, introduced in the Phase 2 development, is a capability to deliver forward-link products to the orbiter ground data system on behalf of the lander. Lander sequence files are "staged" in a local file repository on the host server, with the decision to deliver the files to a specific orbiter made at a later time by the lander operators.

H. Orbiter Forward-Link Repositories

The orbiter forward-link repositories reside outside the boundaries of the system but are an important component of the end-to-end relay process. MaROS provides a streamlined interface to the orbiter project database and handles the delivery of lander sequence files and associated meta-data on behalf of the lander. For the legacy Mars Reconnaissance Orbiter (MRO) and Odyssey missions, the orbiter collections are managed in a system called the Distributed Object Manager (DOM) that is used for many types of flight data at JPL.

III. Relay Services

MaROS provides a set of services in support of relay operations from long-range planning through post-pass analysis. Users interact with the service through the various upload, download, and visualization functions to build an integrated relay plan and report the behavior of the relay network. It is the centralization of services and the correlation of data throughout the planning and execution timeline that provides for end-to-end relay accountability.

A. Long Range Planning

Certain information changes very little over the lifetime of relay and thus can be provided to the system far in advance of the strategic relay planning cycle. Information such as the light time to and from Mars can be calculated years in advance and thus is uploaded very rarely. Other information of this sort includes relay data processing times, which vary from orbiter to orbiter but rarely change for a specific orbiter. This type of information is often correlated with strategic data to calculate forward and return-link pass latencies. The standard MaROS upload services support the provision of all of this data to the system, with back-end latency computation services deriving and persisting the calculated values.

B. Strategic Operations

MaROS provides several services in support of strategic relay operations. Lander and orbiter teams upload and download desired planning information, and make use of the web user interface and command line tools to visualize the relay planning timeline. As each data set is uploaded, derived conflicts and latencies are calculated and recalculated with the latest data provided.

The following describes a fairly nominal strategic scenario:

- Early in the strategic cycle, an orbiter team uploads an Orbiter Sequence of Events (OSOE) file containing a set of “orbiter events”, including timing information such as deep space antenna uplink and downlink windows and associated data rates. Much of this information is incorporated into the calculation of planning warnings, conflicts, and latencies.
- Additionally, the orbiter team uploads the Contact Schedule File (CSF) containing the support schedule for the spacecraft command and control operators. This information also feeds the latency calculations to determine “nominal” forward-link latencies, times of which are useful to the lander teams to know how far in advance of a given relay opportunity any forward-link data must be provided to the orbiter.
- The lander team begins their part of the cycle by uploading the Lander Orbit Propagation and Timing Geometry (LOPTG) file, which defines upcoming pass geometries. At this point there is enough information in the system for the lander team to make some decisions as to which relay session opportunities to request for support in the current strategic planning period. Key geometry parameters such as the geometric view period start time, duration, maximum elevation; and the calculated pass warnings and geometric latencies can be downloaded in an Overflight Summary File (OSF) and utilized in the construction of an Orbiter Request File (ORF), which is the primary mechanism for lander teams to request relay support for their lander.
- Pass decision-making is dependent upon a number of factors. The overflight geometry is an important factor in pass selection. For example, for low-altitude relay orbiters, high pass elevations and longer duration passes are more desirable than lower pass elevations and shorter duration passes. Figure 5 is a screenshot from the web UI depicting two passes with different characteristics, with the blue and green boxes indicating that a pass has been requested.

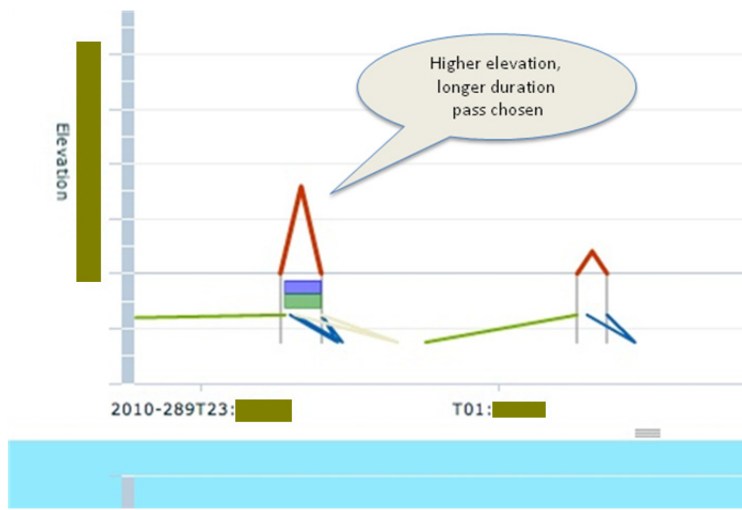

Figure 5. Pass Selection. *The red line represents elevation and time of the pass. The blue and green boxes indicate a pass has been requested.*

- Planning warnings and latencies provide further information for pass planning. These warnings indicate situations where requests could lead to states where relay may not be provided in the manner requested. For example, one orbiter may be in view of two landers at the same time, but is only capable of providing relay for one. This would be indicated as a planning warning, which may be a reason to avoid the overflight and choose another. Latencies provide information regarding uplink and downlink times, and may influence pass decision making as well. For example, data may be desired to be returned by a certain day or time of day for timely delivery to lander science and engineering teams, and thus a pass with an earlier return-link time may be desired over one that ends after the primary operations shift.
- Once the lander team has requested a set of passes for relay the information is submitted to the system as an ORF. This triggers a new set of calculations of latencies based off of the actual requested times as well as any conflicts that may be introduced.
- Once the ORF has been uploaded, notifications are sent to subscribed lander and orbiter team members. The orbiter team then downloads a summary file that correlates the lander’s request information with the original pass geometries, updated pass latencies, and any identified conflicts. As long as there are no conflicts identified, the orbiter team typically responds with a set of acknowledgements that mirror the lander’s requests. However, situations such as orbiter emergencies do arise which occasionally necessitate the orbiter refusing requests or proposing significant request changes. Whatever the orbiter decides, the final acknowledgement is uploaded an Orbiter Acknowledgement File (OAF).

At a certain point in the cycle, the orbiter team implements a relay request as part of it's own sequencing process, and from this point on most orbiters can only make very limited changes to relay plans. This implementation concludes the strategic cycle.

C. Tactical Operations

Once the orbiter has implemented a relay pass, the process enters the "tactical" phase. Changes to relay parameters such as pass duration and data rate may still be modified, potentially as late as the last geometric uplink opportunity between a deep space tracking antenna and the orbiter. However the types of tactical changes that can be made are limited and vary from orbiter to orbiter. For example, one orbiter may be able change the pass start time and shorten or lengthen the pass duration, while another may only be capable of making a pass shorter. As part of the phase 3 development, MaROS is implementing a set of services that allow lander teams to request tactical changes in a manner that is restricted to the capabilities of the orbiter systems, and further provide the orbiters the ability to set what may or may not be changed tactically on an orbiter-by-orbiter basis.

D. Forward-Link Data Management

MaROS provides a set of services in support of lander delivery of sequence file data to the orbiters for relay on a designated pass. MaROS provides a set of repositories for lander missions to stage sequence files for later relay and the tools to manage these files (upload, delete, etc.). Additional services allow the lander user to designate a specific orbiter and pass for files to be delivered through. MaROS tracks a number of parameters related to forward-link such as the sequence file size and the total amount of data delivered for a pass, which may exceed some orbiter limits for data delivery. Additionally, the system provides feedback to the user in the case that the time of delivery exceeds some forward-link latency, meaning that there is a risk the file may not be delivered on the desired pass.

E. Post-Pass Information

Once a pass has been completed the lander and orbiter teams submit analysis data that detail the observed characteristics of the pass. There are two flavors of these reports, a Scorecard file that summarizes key pass values such as the total volume of data received, and an Overflight Performance Assessment File (OPAF) that contains series of data describing any number of measured values over time (power levels, data received, etc.). Both the Scorecard and OPAF data is available for viewing on the web UI and may be used for short or long-term analysis tasks.

F. System Administration and Mission Management

1. *Relay Coordinator/System Administrator*

MaROS provides an interface to the service administrator, which can be used to manage the missions that are included within the relay network. This administration functionality includes capabilities to:

- Add new missions to the network, with the necessary data stores and publication and data extraction privileges.
- Associate "pairs" of lander and orbiters to indicate valid relay partners.
- Enable and disable missions as the network evolves.
- Assign a unique spacecraft identifier for each mission.
- Appoint Mission Administrators for active relay partners.

2. *Mission Administrator*

Each mission has one or more Mission Administrators assigned to them. These individuals are responsible for properly configuring the MaROS services to reflect the capabilities and attributes of their own spacecraft, including capabilities to:

- Add new users for the mission.
- Assign users specific file upload privileges.
- Manage user notification settings.
- Manage other lander or orbiter settings, such as designating the capabilities of an orbiter for managing forward-link products.

3. *Mission User*

Most of the users of the system are considered "Mission Users", who are granted permission to upload different types of data and may also subscribe to notifications that are published as the result of many different kinds of system events.

IV. Conclusion

The transition process from the legacy relay coordination process to MaROS has been met with challenges and in some cases resistance. However, both lander and orbiter mission operators recognize the shortcomings of the legacy system, most notably that introducing a new spacecraft into the Mars Network increases relay complexity by an exponential factor. As such, the operations teams from the ongoing missions at Mars have been generous with their time to both help plan and test a new system that will meet the future challenges of both NASA and international Mars exploration. We have been able to ease the pain of a new system by slowly introducing targeted functionality in various phases of deployment, as the legacy system is retired. Throughout this development and deployment process we have followed a few guiding principles:

- Ensure ubiquitous access through ReSTful and web interfaces;
- Design a system that is mission and even planet agnostic so that future missions may be added with little hassle, and the system itself may be redeployed for other planetary relay networks;
- Accept constant input and feedback between mission operators and the development team to ensure that there is a useful product that may be used for years to come.

We are very proud of the system to date, and look forward to the continued development.

Acknowledgement

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] Hohpe and Woolf, Enterprise Integration Patterns: Designing, Building and Deploying Messaging Solutions, 2004 Addison-Wesley.

[2] Noelios, ReSTlet, http://www.restlet.org

[3] Roy T. Fielding and Richard N. Taylor. 2002. Principled design of the modern Web architecture. ACM Trans. Internet Technol. 2, 2 (May 2002), 115-150. DOI=10.1145/514183.514185 http://doi.acm.org/10.1145/514183.514185

[4] Gladden, R., et al, "Coordinating UHF Relay Activities at Mars", IAA International Conference on Low-Cost Planetary Missions, Noordwijk, The Netherlands, September 2003.

[5] Gladden, R., et al, "Mars Relay Coordination Lessons Learned", IEEE Aerospace Conference #1065, Big Sky, MT, March 2004.

[6] Edwards, C. D., "Relay communications for Mars exploration", International Journal of Satellite Communications and Networking, 2007, 25:111-145.

[7] Gladden, R., "Mars Reconnaissance Orbiter: The History of Supporting the Phoenix Lander", AIAA Space 2009 Conference & Exposition, Pasadena, CA, September 2009.

[8] Edwards, C. D., et al, "Telecommunications Relay Support of the Mars Phoenix Lander Mission", 2010 IEEE Aerospace Conference, Big Sky, MT, March 2010.

[9] Gladden, R., "Mars Relay Operations Service (MaROS): Rationale and Approach", AIAA SpaceOps 2010 Conference, Huntsville, AL, April 2010.

[10] Allard, D., "Strategic Relay Planning and Coordination Interfaces for the Mars Relay Operations Service (MaROS)", 2010 IEEE Aerospace Conference, Big Sky, MT, March 2010.

[11] Allard, D., "Implementation of a Relay Coordination System for the Mars Network", 2010 IEEE Aerospace Conference, Big Sky, MT, March 2010.